Color Image Watermarking and Self-recovery Based on Independent Component Analysis

Hanane Mirza, Hien Thai, and Zensho Nakao

Department of Electrical and Electronics Engineering, University of the Ryukyus, Okinawa 903-0213, Japan
{hanane,tdhien,nakao}@augusta.eee.u-ryukyu.ac.jp

Abstract. The digital image watermarking field addresses the problem of digital image authentication and integrity. In this paper we propose a novel color image watermarking scheme based on image self-embedding and self-recovery techniques. The main idea of this algorithm is to embed a reduced content of the original image to itself, in order to be able to partially recover the deleted features from the watermarked image. Separately, the red and blue color channels are embedded, respectively in the wavelet domain by a compressed version of the original image, and in the spatial domain by binary encoded sequences generated from the original image. This allowed us, in detection stage, to prove the ownership, detect the altered blocks, and recover them. The detection and recovery bits extraction is computed using an ICA algorithm. The experimental results were satisfactory and show a high robustness against most common attacks as well as a reassuring rate of image recovery.

Keywords: Digital watermarking, color image, self-recovery, ICA.

1 Introduction

It is important to know that many court rooms around the world are, nowadays, admitting digital images as legal evidence[4], and many others are seriously considering the digital images for law enforcement[3]. Therefore, the authentication and the integrity of digital images become a serious issue[5], as it can be the key element in a judicial process. On the other hand, the powerful publicly available image processing softwares make digital forgeries very accessible. It is simple for anyone to alter the content of a digital image, by adding or deleting features from the original image without causing detectable edges[9]. In consequence, the new information marketplace, where the digital data can be a currency with two sides, addresses the need and the necessity to produce the softwares and tools necessary to protect the digital images ownership rights, their authentication, and make it possible to recover their original content in case a cutting/pasting attack was performed. Thus, we are proposing in this paper, a new content-based color image self-embedding and self-recovery scheme.

Several algorithms were previously presented regarding this matter, In one of the first techniques used for image tampering detection, Walton *et al* [1],

L. Rutkowski et al. (Eds.): ICAISC 2008, LNAI 5097, pp. 839–849, 2008.

created the theory of check-sums technique, by modifying the Least Significant Bits (LSB) of each pixel. This technique presents a high probability of tamper detecting but it is vulnerable to the block swap attacks. Fridrich *et al* [2] proposed an original fragile watermarking method of image self-embedding that consists of embedding the reduced bits of a block in a different distant block. This method could achieve the self-recovery, but if a distinct region of one image were attacked, the recovery bits would also be corrupted.

In this paper we will try to achieve both objectives, by performing two separate watermarking schemes in two different color layers of the original RGB image. As we have discussed in [6], a color RGB image can be watermarked in its three different color layers (Red,Green and Blue), and the separate watermarking process increases the overall watermarking capacity. In the current paper, we will watermark the red and blue color channels separately, using independent techniques, in order to increase security and the green layer we will keep as original in order not to degrade the quality of the image. In the blue subimage we will try to insert the necessary data to prove the image ownership and to detect its tampered regions. This can be done by embedding the robust bit extracted from a gray level version of the original image, and the embedding process was performed in the spatial domain. As for the red subimage, we inserted a compressed version of the original image, using a DCT compression technique, the embedding is performed in the wavelet domain. The objective of watermarking two color channels with different algorithms is to maximize the chances of content recovery of the image, especially after cutting/pasting attacks. The watermarked image is produced by the superposition of the watermarked red and blue subimage and the unwatermarked green subimage.

As for the extraction stage, the ICA algorithm proved to be an efficient tool[7][8] to detect and extract the mixed unknown sources. It is largely applied for the image and signal processing purposes, and we chose to implement for this experiment the FastICA algorithm for its properties, discussed later. The paper is organized as follows: the second section will discuss the proposed algorithm where we will show how we generate the watermarks to be from the original image separately for both red and blue subimages. Also we will include the embedding process illustrated by figures. In the third section using the chosen ICA algorithm we will demonstrate the detection/extraction process, and the tampered image restoration process. The last two sections will be dedicated to computer simulation results and some of our conclusions.

2 Proposed Algorithm

In the present embedding algorithm (fig.1), we first, separate the original color image $I(N \times N)$ to three color RGB channels, respectively, $R(N \times N)$, $G(N \times N)$ and $B(N \times N)$. For a high embedding capacity[10]we separately watermark the red and blue color channels with two different sets of watermark data that we call R_w, B_w. In the first step we need to generate the watermark data which is

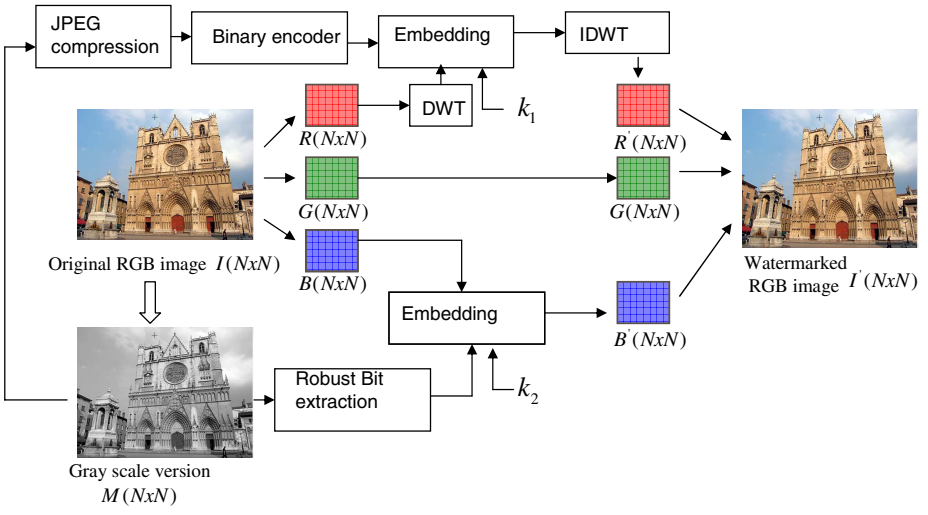


Fig. 1. Embedding process of color layers

content related to the original image, and we leave the green layer untouched, so as not to degrade the invisibility of the final watermarked image.

2.1 Recovery Bits Generation and Embedding

Since the main objective of this proposal is to recover the tampered or deleted areas of the image, we need to generate original image's content related data sets: R_w and B_w. The procedure is as follows:

First, we convert the copy of $I(N \times N)$ to gray-scale level, and we call it $M(N \times N)$.

R_w generation: As for the Red layer $R(N \times N)$ we tried to generate the R_w and perform the embedding in the wavelet domain following the next steps:
1. The Gray-scale image $M(N \times N)$ is divided into 8x8 blocks.
2. Apply $2D\ DCT$ for each block, and divide the entire image by 8 to normalize the DCT coefficients.
3. Quantize the resulting values using the quantization matrix equivalent to the standard *50 %* quality JPEG.
4. The quantized values are further binary encoded using 64 bits only.
5. The resulting binary sequences generated in the previous step can be written as $R_w = \{R_{w1}, ..., R_{wm}\}$, where m is the number of the selected blocks to be compressed.

R_w embedding: On one hand, the red layer $R(N \times N)$ is decomposed into three levels by wavelet transform and the binary sequences R_w are inserted into midfrequency subbands by modifying wavelet coefficients belonging to two details bands at third level (R_3^{LH}, R_3^{HL}). The choice of the embedded wavelet

frequencies is made based on an optimal compromise among robustness, invisibility and attack. The embedding equations are :

$$\begin{aligned} R_3^{'LH}(i,j) &= R_3^{LH} + \alpha_i.R_w(i,j) + \beta.k_1 \\ R_3^{'HL}(i,j) &= R_3^{HL} + \alpha_i.R_w(i,j) + \beta.k_1 \end{aligned} \tag{1}$$

where α is a strength factor adapted to each subband depending on the smoothness and invisibility level, and k_1 is the secret key, containing the block references from which the the binary sequence $R_w$was generated and the subband host order. The watermarked Red layer $R^{'}(N \times N)$ is obtained by applying the inverse DWT.

The main advantage is that we could insert a good amount of image content secretly without changing its perceptual quality, and the the embedded data can be used to recover most of the possible feature deletion.

To apply ICA for watermark extraction algorithm for $R^{'}(N \times N)$, the embedding process needs to create an ICA initialization parameters that we call a $demix\_key$, and we denote D_k_1, calculated by the following equations:

$$D\_k_1 = R_3^{*LH}(i,j) + R_3^{*HL}(i,j) + \gamma k_1(i,j) \tag{2}$$

where R_3^{*LH} and R_3^{*HL} are the subbands coefficient where the key κ is inserted and γ is a mixture strength coefficient, set to 0.5.

B_w generation: The watermark used here is designed in a way to extract a bit sequence of length L containing the robust bits of selected pixels. We divide the image $M(N \times N)$ to $(m \times m)$ block size, using the robust bit extraction algorithm detailed in [11] and illustrated in (fig.2), we extract, from each block, a binary string B_w of fixed length L; $B_w = \sum_{i=1}^{L} B_{wi}$.

B_w embedding: In the other hand, we divide the blue color subimage $B(N \times N)$ into 8x8 blocks.

1. For each block b, we denote the 64 pixels as $P_i \epsilon \{P_1, P_2 ... P_{64}\}$ and we define in a secret key, k_2, the information about each pixel's location in the block b and the block number ($b\#$).
2. The relation between the pixels contained in the block b from the original $B(N \times N)$ and the binary sequences B_{wi} extracted from $M(N \times N)$ is developed according to the following equation:

$$R_i = \Sigma_{i=1}^{64}.B_{wi}.C(P_i) \tag{3}$$

where $C(P_i)$ is the blue color intensity level of the pixel P_i.

3. We encrypt the binary form of the relation R_i and we embed it in the least significant bit of the pixel P_i . We can describe the embedding formula as:

$$P_i^{'} = P_i + R_{ei} + k_2 \tag{4}$$

where R_{ei} is the binary and encrypted form of R_i and $P^{'}$is the watermarked pixel if we denote a watermarked block by $b^{'}(i,j)$, the watermarked blue layer is retrieved by the union of the watermarked blocks:

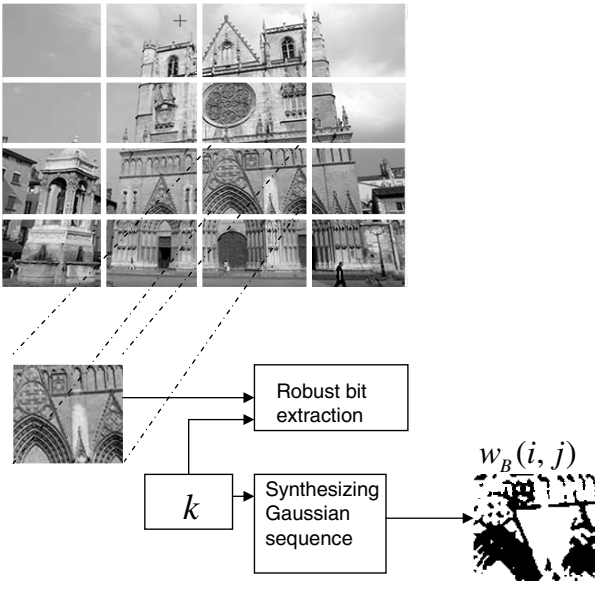


Fig. 2. Robust bits extraction

$$B^{'}(N \times N) = \bigcup_{i=1}^{\frac{N}{8}} \bigcup_{j=1}^{\frac{N}{8}} b^{'}(i,j). \tag{5}$$

The advantage of creating a relation between the extracted sequences and embedded pixels is to make alteration tamper easily detectable and the change in the quality of the image is not noticeable. The relation R_e is block dependent to avoid the loss of embedded data in case the block was removed, as it contains the block # and pixel's location information. Furthermore it is impossible, with this technique, to duplicate an entire block without making undetected damage. The main advantage of this embedding method is that the imperceptibility of the original image is not degraded as we are modifying only the LSB of selected pixels.

Similarly with the red channel, we need to create a demix_key for detection and extraction purposes of this algorithm. The demix_key can be written as:

$$D\_k_2 = P_i^* + k_2 \tag{6}$$

where P_i^* denote the pixels that contain the key k_2.

The watermarked color image $I^{'}(N \times N)$ is obtained by the superposition of the three resulting color layers, $R^{'}(N \times N)$, $B^{'}(N \times N)$ and the green $G(N \times N)$ non-watermarked layer.

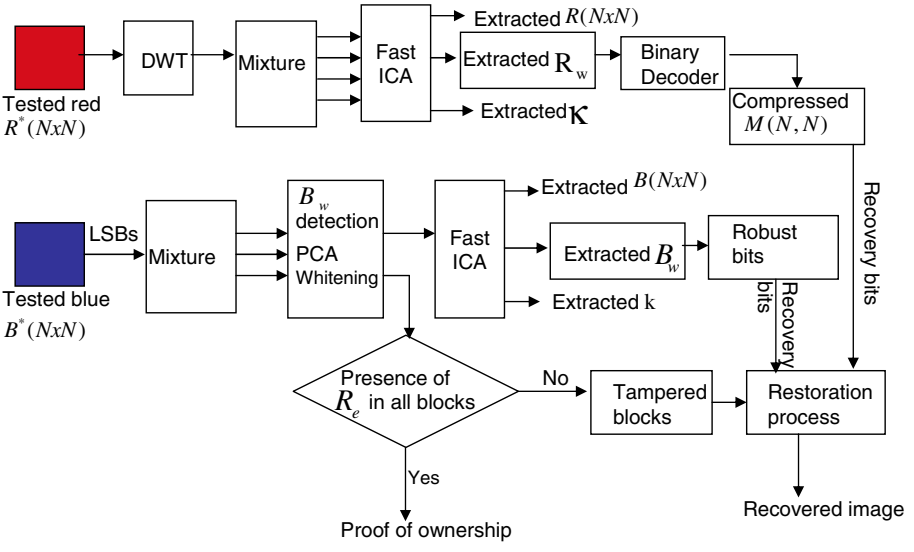


Fig. 3. Proposed detection/extraction process

2.2 Recovery Bits Extraction

Fast ICA

Independent Component Analysis (ICA) is a computing tool to extract independent sources from given mixtures of unknown sources, and this intelligent computing tool is widely applied in signal and image processing field. We consider here that the embedded recovery bits and the original image layers are unknown sources and the watermarked color layers are mixtures of those unknown sources. By creating different mixture the ICA algorithm detects and extracts the embedded recovery bits. Among the presented fixed-point algorithms, we chose in this paper to apply the FastICA [12] because it presents a certain amount of good properties mainly the fast convergence and its easy and suitable implementation for watermarking schemes.

The FastICA is based on two stages: the first is a PCA whitening of the input mixtures and the second is the FastICA by using fourth-order statistics of the signal. The extraction of the recovery bit in the red and blue color channels does not require any knowledge of original embedded recovery bits, or original image, or strength factors. We follow the next steps for extraction process(fig.3):

Step1: The watermarked color image $I^{'}(N \times N)$ is divided to the three color layers $R^{'}(N \times N)$,$G(N \times N)$ and $B^{'}(N \times N)$, We separately apply the FastICA algorithm to both channels $R^{'}(N \times N)$ and $B^{'}(N \times N)$. respectively in step 2 and step 3.

Step2: The extraction process from the red color channel $R^{'}(N \times N)$, using the ICA algorithm is described in the following tasks:

1. The watermarked red layer$R^{'}$ is decomposed through DWT by three levels to obtain wavelet coefficients at $R_3^{\#LH}$ and $R_3^{\#HL}$ subbands.
2. In order to input the initialization parameter of the FastICA algorithm we create mixture signalsX_1, X_2, X_3, X_4 from $R_3^{\#LH}$ and $R_3^{\#HL}$ subbands:

$$\begin{aligned} X_1 &= R_3^{\#LH} + D\_k_1 \\ X_2 &= R_3^{\#HL} + D\_k_1 \\ X_3 &= R_3^{\#LH} + R_3^{\#LH} \\ X_4 &= D\_k_1 + k_1 \end{aligned} \tag{7}$$

3. The mixture signals X_1, X_2, X_3, X_4 are also mixtures of the original wavelet transform coefficients of the original red layer (R_3^{LH}, R_3^{HL}), and the binary sequences R_w and the secret key k_1 which can be written as:

$$\begin{aligned} X_1 &= a_{11}R_3^{LH} + a_{12}R_3^{HL} + a_{13}R_w + a_{14}k_1 \\ X_2 &= a_{21}R_3^{LH} + a_{22}R_3^{HL} + a_{23}R_w + a_{24}k_1 \\ X_3 &= a_{31}R_3^{LH} + a_{32}R_3^{HL} + a_{33}R_w + a_{34}k_1 \\ X_4 &= a_{41}R_3^{LH} + a_{42}R_3^{HL} + a_{43}R_w + a_{44}k_1 \end{aligned} \tag{8}$$

where $a(i,j)\epsilon\{a_{11}, ..., a_{44}\}$ is an arbitrary real number.
4. Using the above described mixtures we can extract, using the fastICA algorithm, from the red layer the embedded binary sequence R'_w.

Step 3: We proceed to extract the embedded recovery bit from the $B^{'}(N \times N)$,proceeding as follows:
1. The checking process is similar to embedding process: it consists of comparing for each block the value of Re^*, determined by the pixels of tested images with the original Re embedded in the LSB, by verifying the pixels locations and the block $b\#$.
2. This correlation process is enough to detect the altered blocks and also to claim the ownership of the original color image.
3. To extract the embedded bits from the watermarked blue layer we create the mixtures, similarly to the red layer :

$$\begin{aligned} X_1 &= P_i + D\_k_2 \\ X_2 &= P_i + e.k_2 \\ X_3 &= D\_k_2 + k_2 \end{aligned} \tag{9}$$

where $e = 0.5$ is the mixture strength factor. The same mixtures are included in the original watermarked image as:

$$\begin{aligned} X_1 &= a_{11}P + a_{12}R_e + a_{13}k_2 \\ X_2 &= a_{21}P + a_{22}R_e + a_{23}k_2 \\ X_3 &= a_{31}P + a_{32}R_e + a_{33}k_2 \end{aligned} \tag{10}$$

4. The above mixtures are used as input data for the fastICA algorithm and the the embedded sequence B_w is extracted.

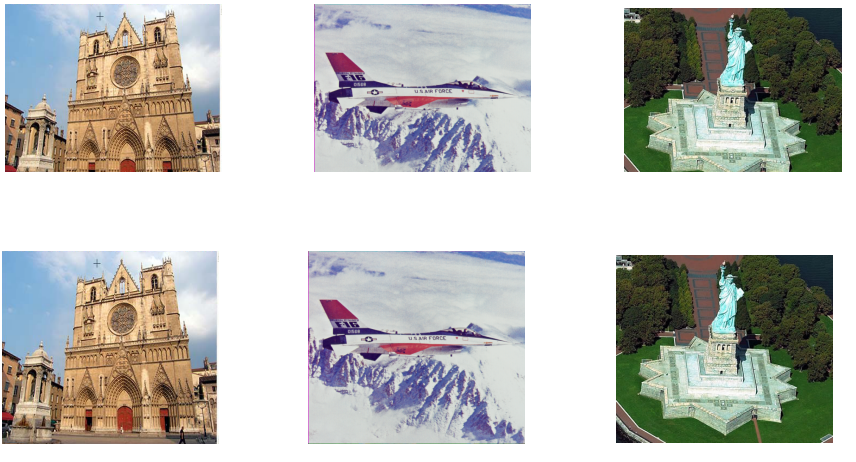
The detected tampered blocks are first restored using the robust bits extracted at this stage, and we recall the extracted content from the red layer to recover each block separately.

3 Computer Simulation

The proposed algorithm was tested on three RGB color images(fig.4(a-c)) of size (512×512) (Cathedral, Airplane and Liberty). The first two images are taken by digital camera while the third one is taken by satellite. The watermarked images are shown in (fig.4(d-f)). No noticeable difference between the original and the watermarked images are detectable for the human eye, and the PSNR values are shown in table (1).

Table(1) shows the PSNR values computed between the original and the watermarked images and the recovery bits detector response in the blue and red channels. The higher is the PSNR, the better is the invisibility of the watermark, and the higher is the detection rate, the better is the robustness.

In order to test the robustness of our algorithm, the watermarked images were subjected to some common image processing attacks, including: Surrounding



(d)Watermarked Cathedral (e)Watermarked Airplane (f)Watermarked Liberty

Fig. 4. Original images (a-c) and Watermarked images(d-f)

Table 1. PSNR values and recovery detector response before the attacks

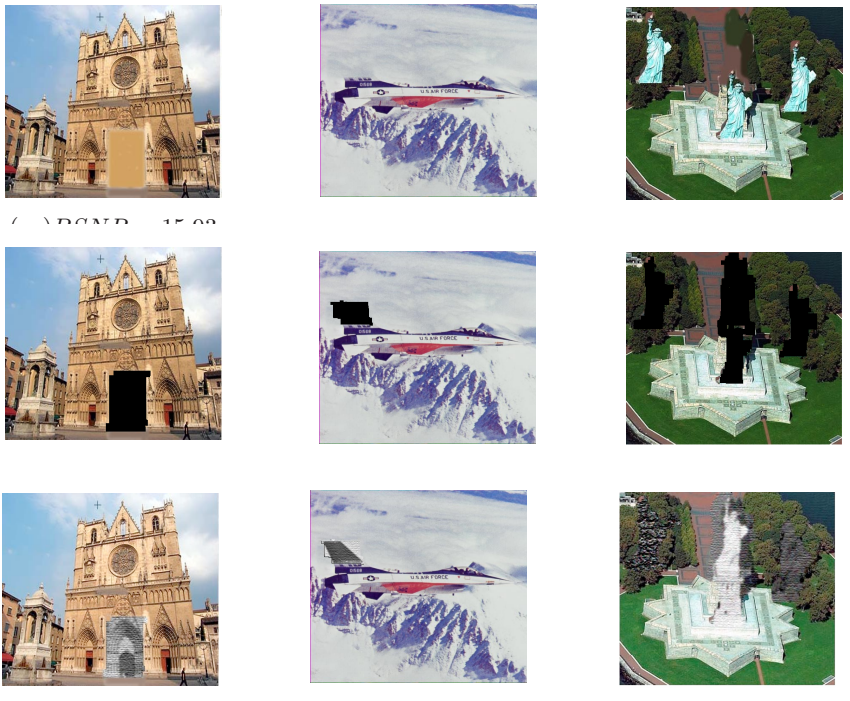
images	(a)	(b)	(c)
PSNR values	42.9	47.0	35.3
B_w detection rate	0.95	0.90	0.91
R_w detection rate	0.93	0.91	0.92

Table 2. Applied attacks and the resulting PSNR values and watermark detector response

Attacks	image(a)			image(b)			image(c)		
	PSNR	XB_w	XR_w	PSNR	XB_w	XR_w	PSNR	XB_w	XR_w
Surrounding Crop (94%)	52.2	0.63	0.68	27.4	0.62	0.58	59.0	0.60	0.54
Resize (448x448)	33.5	0.87	0.79	33.3	0.93	0.89	25.2	0.75	0.72
Adding Noise (power 5000)	17.3	0.81	0.78	15.7	0.87	0.73	15.2	0.76	0.71
Lowpass filtering (3x3)	29.2	0.71	0.63	31.0	0.78	0.67	22.7	0.58	0.63
Median filtering (3x3)	34.2	0.58	0.63	33.7	0.66	0.58	23.1	0.680	0.63
Jpeg (Quality=85%)	33.4	0.76	0.72	34.0	0.73	0.70	27.0	0.71	0.68
Jpeg2000 (bpp = 0.25)	32.7	0.86	0.77	33.3	0.81	0.79	23.9	0.76	0.68

Crop, Resize, Adding Noise, Lowpass filtering, Median filtering, Jpeg, and Jpeg2000. The results are shown in table(2).

After performing the attacks on the watermarked images the PSNR was re-calculated and so was the the Recovery bits extraction: we indicated the amount



$(c_1) PSNR = 21.1$ $(c_2) PSNR = 22.1$ $(c_3) PSNR = 20.5$

Fig. 5. Sample results of the proposed tamper detection and recovery algorithm

of the recovery bits detected in the watermarked tested blue channel as XB_w and the one in the red channel as XR_w.

The system showed good results for the robustness as shown in table (2). The embedded watermark was detectable enough in most tests, to prove ownership of the file at least, if not to also recover partially some of the attacked features.

As for the image restoration testing, the three watermarked images were tested by being subjected to cutting/pasting attacks, and we tried to recover the deleted feature from the extracted recovery bits. Fig.5($a_1 - a_3$) shows the modified watermarked images, the modification purpose was to delete some of the features of the watermarked images: In the Cathedral image (a_1) the middle gate was 'cut' and colored in similar color to the main color of the rest of Cathedral; in the Airplane image (a_2) the tail of the Airplane that carried all the references was deleted; in the Liberty image (a_3) the liberty statue was cut off its original stand and pasted to three different locations in same image (right, left and below the original stand). The PSNR between the tampered images and the original images was computed as well and it is shown in fig.5($a_1 - a_3$). As figure(5)($b_1 - b_3$) shows, the altered blocks of each image were detected by the algorithm. The partially recovered features are shown in fig.5 ($c_1 - c_3$). The tampered images are not perfectly recovered but enough to have an idea about the original features, which guarantee the image authentication, and make it reliable as a legal evidence.

4 Conclusions

The main contribution of this paper is to demonstrate that it is possible, through the theory and computer simulation (more results and comparisons to come), to recover the original content of a tampered color image. Watermarking two color layers is done to double the chances of recovering the embedded data and two different independent algorithms are performed to increase the security and robustness of the algorithm. The ICA extraction algorithm makes it possible to extract the embedded data after all the attacks are performed. The experimental results shows that the watermark survived all the common attacks, and the embedded content-based watermarked were still detectable, extractable and useful for tampered regions recovery.

Acknowledgments

This research was supported in part by Ministry of Internal Affairs and Communications (Japan) under Grant: SCOPE 072311002, for which the authors are grateful.

References

1. Walton, S.: Information authentication for a slippery new age. Dr. Dobbs Journal 20, 18–26 (1995)
2. Fridrich, J., Goljan, M.: Protection of digital images using self embedding. In: Symposium on Content Security and Data Hiding in Digital Media, New Jersey Institute of Technology, USA (1999)

3. Craiger, P.J., Pollitt, M., Swauger, J.: Law enforcement and digital evidence. Handbook of Information Security, New York, USA (2005)
4. Staggs, S.B.: The Admissibility of Digital Photographs in Court (2005), http://www.crime-scene-investigator.net/admissibilityofdigital.html
5. Mason, S.: Authentication of electronic evidence. Information Age, Australia (2006)
6. Miyara, K., Thai, H., Harrak, H., Nakao, Z., Nagata, Y.: Multichannel color image watermarking using PCA eigenimages. In: Advances in Soft Computing, vol. 5, pp. 287–296. Springer, Heidelberg (2006)
7. Yu, D., Sattar, F., Ma, K.-k.: Watermark Detection and Extraction Using Independent Component Analysis Method. Journal on Applied Signal Processing, EURASIP 5, 92–104 (2002)
8. Shen, M., Zhang, X., Sun, L., Beadle, P.J., Chan, F.H.Y.: A method for digital image watermarking using ICA. In: 4th International Symposium on Independent Componenet Analysis and Blind Signal Separation (ICA 2003), Japan (2003)
9. Rehmeyer, J.: Computing Photographic Forgeries. Science News, vol. 171 (2007)
10. Barni, M., Bartolini, F., De Rosa, A., Piva, A.: Color image watermarking in the KLT domain. In: SPIE, Electronic Imaging, pp. 87–95 (2002)
11. Fridrich, J.: Robust Bit Extraction from Images. In: IEEE International Conference on Multimedia Computing and Systems (ICMCS 1999), vol. 2 (1999)
12. Bingham, E., Hyvarinen, A.: A fast fixed-point algorithm for independent component analysis of complex-valued signals. Int. Journal of Neural Systems 10, 1–8 (2000)